UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/957,459	09/21/2001	Warren Roach	166.0001	2504

25534 7590 05/12/2010
CAHN & SAMUELS LLP
1100 17th STREET NW
SUITE 401
WASHINGTON, DC 20036

EXAMINER	
TO, BAOQUOC N	

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
05/12/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 09/957,459	**Applicant(s)** ROACH ET AL.	
	Examiner BAOQUOC N. TO	**Art Unit** 2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _04/17/2010_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-16,18,34-49 and 51-59_ is/are pending in the application.

4a) Of the above claim(s) _17, 19-33 and 50_ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-16,18,34-49 and 51-59_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This communication is responsive to the Appeal Brief, filed 04/17/2009. Claims 1-16, 18, 34-49 and 51-59 are pending in this communication, and this office action is made non-final. This action intends to correct the number of the rejected claims in the Rejections in order to enter the Appeal Brief filed 04/17/2009. This Office Action will include the rejection for claims **52 and 53** which inadvertently left out in the last Office Action. Examiner apologies for any inconveniences which cause to the applicants.

2. Reopening of Prosecution After Appeal Brief or Reply Brief

In view of the Appeal Brief filed on 04/17/2009, PROSECUTION IS HEREBY REOPENED.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:

/John Breene/
Supervisory Patent Examiner, Art Unit 2162

Claims cancelled:	17, 19-33 and 50.
Claims amended:	None.
Claims pending:	1-16, 18, 34-49 and 51-59.

Response to Arguments

3. Applicant's arguments filed 04/17/2009 have been fully considered but they are not persuasive.

A. Rejections (a) –(c)

Applicant argues "the Official made an incorrect factual finding that the claim term "operating system" was equivalent to the API of Koshisaka. The Official Action improperly relied upon this factual finding as a basis for its obviousness conclusions in all of the rejections. The Office fundamentally misconstrued the scope and meaning of the term API as used in Koshisaka..."

The examiner respectfully disagrees with the above argument. While applicant's evidences such as Bryan Pfaffenberger, Webster's NEW WORLD, Copyright 2000 and Steven R. William, DECLARATON UNDER 37 CFR 1.132 to distinguish between the Operating System and API are two different and distinct programs ; however, examiner also provided the evidence such the "Application Program Interface" is a part of the

Operating System. Secondly, regarding to the DECLARATION UNDER 37 CFR 1.132 of Steven R. William and the definition of "Operating System" from Bryan Pfaffenberger, Webster's NEW WORD which is not enough sufficient evidence to distinguish that the "Operating System" is not an "API" by definition, the following is the quotation copy exactly from the applicant's provided evidence "Operating system (OS) A master control program that manages the computer's internal function, such as accepting keyboard input, and the provide a means to control the computer's operations and file system." Clearly, this does not provide any evidence supporting the "Operating system" is not "API" or API is not a part of the "operating system" which the examiner provided the evidence for supporting this.

Furthermore, Examiner equates API as to applicant claimed "operating system" which both are the software program used in the computer system to perform command which the user requested. In the broadest interpretation of the "API" and "Operating system", they are both are software process. Furthermore, examiner also provide a second reference showing the monitoring process which were disclosed by Dunphy and Parthsarathy to provide a support that API is a part of the operating system (col. 9, lines 60-63) and the detection is possibly by API by Koshisaka. Moreover, in view of the guidance provided by the Supreme Court in *KSR* decision, the a patent claim is prima facie obvious if "some motivation or suggestion to combine the prior art teachings" can be found in the prior art, the nature of the problem, or the knowledge of a person having ordinary skill in the art. See the recent Board decision *EX parte Smith*, --USPQ2d--, slip op. at 20, (Bd. Pat. App. & Interf. June 25, 2007 (citing *KSR*, 82 USPQ2d at 1396) (available at <http://www.uspto.gov/web/offices/dcom/bpai/prec/fd071925.pdf>).

Applicant also agues “the Office alleges that Parasarathy that its API is part of its operating system. For this assertion, the Office Action relies of Col. 5, lines 59-61. However, the cited passages of Parasarathy address neither operating system nor APIs."

Examiner disagrees with the above argument. The citation which provided by examiner in the Office Action was a typo. It should have cited on col. 9, lines 60-62 which states "the hashing component 260 and the digital signature component 265 are can be an application program interface (API) that resides as part of an operating system 275" (col. 9, lines 60-62). The conclusion is API is a part of the operating system which clearly identified by Parasarathy.

B. Rejection of Claims 1-18 and 54-57 under 35 U.S.C 103(a).

Applicant argues “notwithstanding the comments to the contrary in the Office Action, neither Koshisaka or Dunply, taken alone or in combination, teach or disclose this detection step…”

Examiner respectfully disagrees with the above argument. As previously indicated, Koshisaka and Dunply disclosed a monitor process and capture the process before the actual process being executed. The API as indicated by Koshisaka is "Operating system". Applicant also provided a definition from the Dictionary to distinguish the API and operating system. However, the examiner also provides the

API is a part of the operating system (col. 9, lines 60-63) by Parthsarathy.

Applicant argues “in direct contrast, Koshisaka is not data-centric; that is, it does not teach detection…”

Examiner respectfully disagrees with the above argument. This is not relevant to the claimed invention or anything which is claimed.

Applicant argues “unlike the method of claim 1, neither Dunply nor Parasarathy teach detecting an instructions by an operating system. The portion of Parasarathy that the Office Action replies on for such a teaching does not even mention API or operating system. Thus, Dunphy and Parasarathy suffer from the same deficiencies as Koshisaka.”

Examiner respectfully disagrees with the above argument. As indicated in section A argument. The Parasarathy indicate API is a part of the operating system in which provide support for the API is a part of the operating system.

C. The rejection of claims 34-38, 43-51 and 57-59.

Applicants argues “in addition, neither Dunply, Parasarathy nor Koshisaka teach strong an archive file created from the operating file in a temporary first storage location responsive to detection of the operating system instructions...."

Examiner respectfully disagrees with the above argument. As shown by Koshisaka, after the detection of the command delete from the user. The file is being backup and stored in the memory (col. 6, lines 44-54).

Applicants argues “unlike the method of claim 34, Dunply does not detect an instruction by an operating system. Dunply, such like Koshisaka and other references of record, is concerned with communication from the application program...."

Examiner respectfully disagrees with the above argument. As explained above, the detection was done by the API as discloses by Koshisaka. And furthermore, the cited Parasarathy shows the API as a part of the operating system which perform the detection process.

i. Claim 44

Applicant argues “with respect to claim 44, it requires that the archive file pass through two storage locations before ending up in permanent (its third storage location)..."

The examiner respectfully disagrees with the above argument. The event log as recited in the Koshisaka as being equated to the storage locations.(col. 4, liens 25-38).

D. The Rejection of claims 39-42.

Applicant argues “...At best, Midgely teaches generating a notification when a user requests a file open operation. It does not suggest the claimed operation of searching..."

Examiner respectfully disagrees with the above argument. The passage as indicated by Midgely provides a notification process which notifies the user in one schedule event (col. 7, lines 59-63). This passage provide support for notification of based on the schedule event which is equivalent to the searching a first storage location for the archive file responsive to receipt of a message from a timer.

ii. Claims 41 and 42.

Applicant agues "while the Office Action relied on Midgely for teaching that an agent is notified when a client requests a file open operation prior to executing the open operation, Midgely offers not teaching that is remotely related to the limitation of claims 41-42..."

Examiner respectfully disagrees with the above argument. Midgely discloses a notification process based on the schedule event such as when a client request a file open operation (col. 7, lines 61-64). Such notification is equivalent to the notification to move the archive file to the permanent location from the temporary location.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-16, 18, 34-49 and 51-59 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1, lines 5, recites "capturing the operating file temporally ***proximate*** to the operation on an…” render indefinite because there is no metes and bounds for this language.

Claim 34, in lines 6, recites "....first storage location temporally ***proximate*** to the operation..." render indefinite because there is no metes and bounds for this language.

Claim 54, in line 4, recites “capturing the operating system ***just before or just after*** the operation being performed…” render indefinite because there is no metes and bounds for this language.

Claims 59, in lines 6, recites “…moving the archive file to a first storage device temporally ***proximate*** to the operation being....” render indefinite because there is no metes and bounds for this language.

For the purpose of examination, meaning of “…temporally proximate, just before, or just after” are being interpreted as any time before or any time after the operation being performed. Appropriate correction is required.

35 USC § 101

6. The method as recited in claim(s) 1, 34, 54 and 59 will be interprets and give a specific meaning to the recited method which the methods perform by a computer including a processor and memory having instructions execute by the processor to perform and negating any embodiment which expressly disclose the methods is an mental or abstract idea or method will be perform without a computer and any other methods do not fall within the definition of 35 U.S.C 101

The computer usable medium as claimed will be interpreted as storage medium.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-16, 18 and 52-57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Koshisaka (US. Patent No. 6,629,109 B1) in view of Dunphy et al. (US. Patent 5,638,509) and further in view of Parthasarathy et al. (US. Patent No. 7,117,371 B1).

Regarding on claim 1, Koshisaka teaches in computing device, a method for archiving files comprising:

Detecting an instruction by an operating system to perform an operation on an operating file (the file manipulation monitoring section 21 of the file revision management system 2 detects file manipulation which is going to be executed by the application 1 step Si…) (col. 6, lines 32-43); and

Although, Koshisaka does not explicitly teach capturing the operating file temporally proximate to the operation being performed on the operating file, responsive to the detection of the instruction. Koshisaka does not explicitly teach capturing the operating file temporally proximate to the operation being performed on the operating file, responsive to the detection of the instruction. However, Koshisaka teaches “the file manipulation monitoring section 201 constantly monitors API (Application Program Interface) commands which are outputted by the application 1 to the operating system 3 and thereby detects the file manipulation which is (going to be) executed by the application 1…In the case wherein the file manipulating is “file deletion” (“Yes” in the step S2), the file manipulation monitoring section 21 instructs the processing section 22 to store a “deleted file name” and a corresponding “backup file name” in the deleted file name memory section 23 (step 23). The deleted file name is the name of the file (to be) deleted by the application 1.” (col. 6, lines 35-54). This teaches the monitoring program send out the instruction to save the deleted file to the backup memory right at the time the program can execute the delete application. On the other hand, Dunphy also discloses capturing the operating file temporally proximate to the operation being performed on the operating file, responsive to the detection of the instruction (col. 3, lines 50-67 and col. 4, lines 1-10). Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka’s system to include storing the original file in the memory as taught in Koshisaka in order to allow the user to retrieve latter on when needed. Further more, the both of Koshisaka and Dunphy discloses detecting is done by an API and API is not a part of the Operating System. On the other hand, Parthasarathy discloses an API is a part of the Operating System (col. 5, lines 59-61). Since, API in Parthasarathy disclosed as a part of the operating system. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify the API of as disclosed in Parthasarathy to detect an instruction performing on as file as Koshisaka and Dunphy to generate a backup for later recovery.

Regarding on claim 2, Koshisaka teaches capturing the operating file includes creating an archive file and storing the archive file in a storage location (col. 6, lines 35-45).

Regarding on claim 3, Koshisaka teaches the archive file includes copy of the operating file (col. 6, lines 35-45).

Regarding on claim 4, Koshisaka does not explicitly teach the archive files includes portions of the operating file. However, Dunphy teaches "the data file monitor 11 creates entries in event log 12 that identifies the data directory/data file, the nature of the change, extent of the data file, the time that this change occurred and any other pertinent administrative information…" (col. 4, lines 33-37). This suggests the event log creating entry containing the changes of the file. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka's system to include the entry containing changes to the file as taught in Dunphy in order to provide the system retrieve and restore partial of the file in the event of users change their mind.

Regarding on claim 5, Koshisaka does not explicitly teach the archive file includes pointers directed to one or more storage locations, wherein each of the one or more second storage locations contains at least a portion of the operating file. However, Dunphy teaches "a database 14 located in the data storage and protection apparatus 10 retrieves the event log 12 and uses the information contain therein to identify data files that are to be transmitted to a data file backup media 21 for storage. The database also contains a complete history of all data file changes since it stores the event log entries in its history file" (col. 4, lines 41-46). This suggests the database 14 is the second storage having different entries of a file. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka's system to include the entry containing changes to the file as taught by Dunphy in order to provide the system retrieve and restore partials of the file from the event log in the event of users change their mind.

Regarding on claim 6, Koshisaka does not explicitly teach capturing the file includes saving the archive file prior to the operation being performed on the operating file. However, Dunphy teaches "the data file monitor 11 creates entries in event log 12 that identifies the data directory/data file, the nature of the change, extent of the data file, the time that this change occurred and any other pertinent administrative information…" (col. 4, lines 33-37). This suggests the event log creating entry containing the changes of the file, is the saving the archive file prior the executing the instruction. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka's system to include the entry containing changes to the file as taught in Dunphy in order to provide the system retrieve and restore partial of the file in the event of users change their mind.

Regarding on claim 7, Koshisaka does not explicitly teach the file includes saving the archive file subsequent to detecting the instruction to perform the operation. However, Dunphy teaches “the data file monitor 11 creates entries in event log 12 that identifies the data directory/data file, the nature of the change, extent of the data file, the time that this change occurred and any other pertinent administrative information…” (col. 4, lines 33-37). This suggests the event log creating entry containing the changes of the file, is the saving the archive file prior the executing the instruction. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka’s system to include the entry containing changes to the file as taught in Dunphy in order to provide the system retrieve and restore partial of the file in the event of users change their mind.

Regarding on claim 8, Koshisaka does not explicitly teaches capturing the file includes saving the archive file subsequent to the operation being performed on the operating file. Dunphy teaches “the data file monitor 11 creates entries in event log 12 that identifies the data directory/data file, the nature of the change, extent of the data file, the time that this change occurred and any other pertinent administrative information…” (col. 4, lines 33-37). This suggests the event log creating entry containing the changes of the file, is the saving the archive file prior the executing the instruction. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka’s system to include the entry containing changes to the file as taught in Dunphy in order to provide the system retrieve and restore partial of the file in the event of users change their mind.

Regarding on claim 9, Koshisaka teaches the storage location includes a buffer (col. 5, lines 55-65).

Regarding on claim 10, Koshisaka teaches the first storage location includes a storage device (col. 6, lines 32-65).

Regarding on claim 11, Koshisaka teaches the storage device includes at least one of a group comprising a magnetic storage medium, an optical storage medium, and a solid state storage device (col. 6, lines 32-65).

Regarding on claim 12, Koshisaka teaches the storage location includes a directory disposed on said storage device (col. 6, lines 32-65).

Regarding on claim 13, Koshisaka doest not explicitly teach determining whether the operating file has previously been captured prior to capturing the file. However, Dunphy teaches "if a data change is detected, at step 34, the data file monitor 11 extracts data file status activity information from the received communications and uses this data to maintain an event log 12 indicate a history of all presently occurring data file activity on the computer system1..." (col. 3, lines 64-67 to col. 4, lines 1-21). This suggests that the event log 12 storing the file prior the change made to the file. Therefore, would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Koshisaka's system to include the entry containing changes to the file as taught in Dunphy in order to provide the system retrieve and restore partial of the file in the event of users change their mind.

Regarding on claim 14, Koshisaka discloses determining whether the operating file has previously been captured prior to capturing the file (col. 6, lines 32-65).

Regarding on claim 15, Koshisaka teaches the operation causes a change in the operating file (col. 5, lines 32-65).

Regarding to claim 16, Koshisaka discloses an article of manufacture comprising a computer usable medium having computer program code for performing of the method of claim 1 (an storage unit such as memory, an HDD (Hard Disk Drive), etc) (col. 5, lines 60-62).

17. (Cancelled).

Regarding to claim 18, Koshisaka discloses an article manufacture comprising a processor configured to perform the method of claim 1 (an storage unit such as memory, an HDD (Hard Disk Drive), etc) (col. 5, lines 60-62).

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Cancelled)

26. (Cancelled)

27. (Cancelled)

28. (Cancelled)

29. (Cancelled)

30. (Cancelled)

31. (Cancelled)

32. (Cancelled)

33. (Cancelled)

As to claim 52, Koshisaka discloses the method of claim 2, wherein said capturing step occurs only if a match to a defined condition has been determined (the condition is "file deletion") (col. 6, lines 44-67).

As to claim 53, Koshisaka discloses the method of claim 52, wherein said defined condition includes at least one of determining whether the operating file has previously been archive (after the file saved and file manipulation such as changing the file name which trigger file deletion and backup) (col. 6, lines 5-67) and determining whether the operating file has been selected for protection.

Claim 54 is rejected same reason as claim 1, furthermore, Koshisaka the command is intercepted by the API prior to execute the command (col. 6, lines 32-43).

Regarding on claim 55, Koshisaka teaches the method recited in claim 54, wherein said capturing occurs an instant before (the file manipulation monitoring section 21 of the file revision management system 2 detects file manipulation which is going to executed by the application 1 step Si) (col. 6, lines 32-35) or after the operation is performed on the operating file.

Regarding on claim 56, Koshisaka does not explicitly teach the method recited in claim 54, wherein the operating file is a system file (file system) (col. 10, lines 38-42).

Regarding on claim 57, Koshisaka teaches the method recited in claim 54, wherein the operating file is a user file (user file) (col. 6, lines 5-43).

8. Claims 34-38, 43-49, 51 and 58-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunphy (US. Patent No. 5,638,509) further in view Koshisaka (US. Patent No. 6,629,109 B1) and further in view of Parthasarathy et al. (US. Patent No. 7,117,371 B1).

Regarding on claim 34, Dunphy teaches in a computing device, a method for archiving files comprising:

Detecting an instruction by an operating system to perform an operation on an operating file (application program resident on the computer system to intercept all communication therebetween) (col. 1, lines 60-61);

Searching the first temporary storage location for the archive file responsive to the occurrence of the fist event (the database 14 located in the data storage and protection apparatus 10 retrieves the event log 2) (col.4, lines 40-42); and

Moving the archive to a second storage location responsive to a second event, the second storage location being a permanent storage location (uses the information contained therein to identify data files that are to be transmitted to data file backup media 21 for storage) (col. 4, lines 43-45).

Dunphy does not explicitly creating an archive file from the operating file and storing the archive file in a temporary first storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction. However, Dunphy teaches "if a data file change is detected, at step 34, the data file monitor 11 extracts data file status and activity information from the received communications an uses this data to maintain an event log 12 that indicates a history of all presently occurring, data file activity on the computer system 1…" (col. 3, lines 49-67 to col. 4, lines 1-21). In addition, Dunphy teaches "the data file monitor 11 creates an entry in event log 2 that identifies the data directory/data file the nature of the change, extent of the data file, the time that is change occurred and any other pertinent administrative information, such as a user identification, that may be pertinent to the operation of the data file storage and protection system 10" (col. 4, lines 33-38). On the other hand, Koshisaka teaches "the file manipulation monitoring section 201 constantly monitors API (Application Program Interface) commands which are outputted by the application 1 to the operating system 3 and thereby detects the file manipulation which is (going to be) executed by the application 1…In the case wherein the file manipulating is "file deletion" ("Yes" in the step S2), the file manipulation monitoring section 21 instructs the processing section 22 to store a "deleted file name" and a corresponding "backup file name" in the deleted file name memory section 23 (step 23). The deleted file name is the name of the file (to be) deleted by the application 1." (col. 6, lines 35-54). This teaches the monitoring program send out the instruction to save the deleted file to the backup memory right at the time the program can execute the delete application. These functionalities in Koshisaka and the current application are the same. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify the teaching of capturing the delete file to the back up memory as taught in Koshisaka in order to allow the user to retrieve latter on when needed. Further more, the both of Koshisaka and Dunphy discloses detecting is done by an API and API is not a part of the Operating System. On the other hand, Parthasarathy discloses an API is a part of the Operating System (col. 5, lines 59-61). Since, API in Parthasarathy disclosed as a part of the operating system. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify the API of as disclosed in Parthasarathy to detect an instruction performing on as file as Koshisaka and Dunphy to generate a backup for later recovery.

Regarding on claim 35, Dunphy teaches the method recited in claim 34 wherein storing the archive file includes storing the archive file prior to the operation being performed on the operating file (creating a entry in the event log prior executing) (col. 4, lines 30-38).

Regarding on claim 36, Dunphy teaches the method recited in claim 35 wherein storing the archive file includes storing the archive file prior to the operation being performed on the operating file and subsequent to the operation being performed on the operating file (the entries is created based on the change is made to the file and change is made to the file after the intercepting command) (col. 3, lines 36-67).

Regarding on claim 37, Dunphy teaches the method recited in claim 34 wherein storing the archive file includes storing the archive file subsequent to the operation being performed on the operating file (history file) (col. 4, lines 41-46).

Regarding on claim 38, Dunphy teaches the method recited in claim 34 wherein the first temporary storage location includes a buffer (log) (col. 4, lines 24-34).

Regarding on claim 43, Dunphy teaches the method recited in claim 34 wherein the second storage location is an output buffer (col. 4, lines 40-45).

Regarding on claim 44, Dunphy teaches the method recited in claim 34 further comprising:

after storing the archive file in the first temporary storage location, updating a database to indicate that the archive file is located in the first storage location (col. 4, lines 25-67);

Determine a final destination for the archive file (col. 4, lines 25-67);

Moving the archive file from the first temporary location to an intermediate storage location (col. 4, lines 25-67);

Updating the database to indicate that the archive file are located in the intermediate storage location (col. 4, lines 25-67); and

After moving the archive file to the second storage location, updating the database to indicate that the archive file is located in the second storage location (col. 4, lines 25-67).

Regarding on claim 45, Dunphy teaches the method recited in claim 44 wherein the second location include a personal attached storage device (backup disk 21) (col. 4, lines 42-44).

Regarding on claim 46, Dunphy teaches the method recited in claim 45 wherein the second storage location includes a network attached storage device (the backup device 20 can be collocated with computer system 1 or can be located remote from computer system 1 and connected thereto via a data communication) (col. 3, lines 30-34).

Regarding on claim 47, Dunphy teaches the method recited in claim 44 wherein the second storage location includes a peer-to-peer storage device (the backup device 20 can be collocated with computer system 1 or can be located remote from computer system 1 and connected thereto via a data communication) (col. 3, lines 30-34).

Regarding on claim 48, Dunphy teaches the method recited in claim 44 wherein the second storage location includes an Internet storage area network (the backup device 20 can be collocated with computer system 1 or can be located remote from computer system 1 and connected thereto via a data communication) (col. 3, lines 30-34).

Regarding to claim 49, Dunply disclose an article of manufacture comprising a computer usable medium having program code for performing the method of claim 44 (backup media 21) (col. 3, lines 22-35).

Claim 50. (Cancelled)

Regarding to claim 51, Dunply discloses an article of manufacture comprising a processor configured to perform the method of claim 44 (backup media 21) (col. 3, lines 22-35).

Regarding on claim 58, Dunphy teaches the method recited in claim 34 wherein the first event is different from said second event (col. 4, lines 24-50).

Regarding on claim 59, Dunphy teaches in a computing device, a method for archiving files comprising:

Detecting an instruction by operating system to perform an operation on an operating file (application program resident on the computer system to intercept all communication therebetween) (col. 1, lines 60-61); and

Storing the archive file in a second storage device (a database 14 located in the storage and protection apparatus 10 retrieves the event log 12 and uses the information contained therein to identify data files that are to be transmitted to a data file backup media 21 for storage) (col. 4, lines 41-44).

Dunphy does not explicitly teach creating an archive file from the operating file and moving the archive file to a first storage device temporally proximate to the operation being performed on the operating file, responsive to detecting the instructions. However, Dunphy teaches "if a data file change is detected, at step 34, the data file monitor 11 extracts data file status and activity information from the received communications an uses this data to maintain an event log 12 that indicates a history of all presently occurring, data file activity on the computer system 1…) (col. 3, lines 49-67 to col. 4, lines 1-21). In addition, Dunphy also teaches "the data file monitor 11 creates an entry in event log 2 that identifies the data directory/data file the nature of the change, extent of the data file, the time that is change occurred and any other pertinent administrative information, such as a user identification, that may be pertinent to the operation of the data file storage and protection system 10" (col. 4, lines 33-38). This suggests that changed instruction is intercepted and an entry log is created and store in the event log. On the other hand, Koshisaka teaches "the file manipulation monitoring section 201 constantly monitors API (Application Program Interface) commands which are outputted by the application 1 to the operating system 3 and thereby detects the file manipulation which is (going to be) executed by the application 1…In the case wherein the file manipulating is "file deletion" ("Yes" in the step S2), the file manipulation monitoring section 21 instructs the processing section 22 to store a "deleted file name" and a corresponding "backup file name" in the deleted file name memory section 23 (step 23). The deleted file name is the name of the file (to be) deleted by the application 1." (col. 6, lines 35-54). This teaches the monitoring program send out the instruction to save the deleted file to the backup memory right at the time the program can execute the delete application. These functionalities in Koshisaka and the current application are the same. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modifying Dunphy's system to include capturing the delete file to the back up memory as taught in Koshisaka in order to allow the user to retrieve latter on when needed. Further more, the both of Koshisaka and Dunphy discloses detecting is done by an API and API is not a part of the Operating System (col. 5, lines 59-61). On the other hand, Parthasarathy discloses an API is a part of the Operating System. Since, API in Parthasarathy disclosed as a part of the operating system. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify the API of as disclosed in Parthasarathy to detect an instruction performing on as file as Koshisaka and Dunphy to generate a backup for later recovery.

9. Claims 39-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunphy et al. (US. Patent No. 5,638,509) in view Koshisaka (US. Patent No. 6,629,109 B1) and further in view of Parthasarathy et al. (US. Patent No. 7,117,371 B1) and further in view of Midgely et al. (US. Patent No. 5,608,865).

Regarding on claim 39, Dunphy, Koshisaka and Parthasarathy do not explicitly teach the method recited in claim 34 wherein the first event includes message from a timer. However, Midgely teaches "the protected server's protection agent registers with the Netware file system's File system Monitor feature. This registration requests that the agent be notified when a client a requests a file open operation, prior to the file system's execution of the open operation" (col. 7, lines 59-63). This suggests the notification is the message from a timer. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Dunphy and Koshisaka and Parthasarathy system to include notification as taught by Midgely in order to notify the user the changes that is about to be made to the file to allow the user to take the next appropriate actions.

Regarding on claim 40, Dunphy, Koshisaka and Parthasarathy do not explicitly teach the method recited in claim 34 wherein the first event includes a message from a program resident on the computing device. However, Midgely teaches "the protected server's protection agent registers with the Netware file system's File system Monitor feature. This registration requests that the agent be notified when a client a requests a file open operation, prior to the file system's execution of the open operation" (col. 7, lines 59-63). This suggests the resident program is the agent. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Dunphy, Koshisaka and Parthasarathy system to include an agent as taught by Midgely in order to notify the change that is about to be made to the file to allow the user to take the next appropriate actions.

Regarding on claim 41, Dunphy, Koshisaka and Parthasarathy do not explicitly teach the method recited in claim 34 wherein the second event includes a message from a timer. However, Midgely teaches "the protected server's protection agent registers with the Netware file system's File system Monitor feature. This registration requests that the agent be notified when a client a requests a file open operation, prior to the file system's execution of the open operation" (col. 7, lines 59-63). This suggests the same concept of notification system. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Dunphy, Koshisaka and Parthasarathy system to include an notification system as taught by Midgely in order to notify the user take appropriate actions.

Regarding on claim 42, Dunphy, Koshisaka and Parthasarathy do not explicitly teach the method recited in claim 34 wherein the second event includes a message indicating when the second storage location is available. However, Midgely teaches "the protected server's protection agent registers with the Netware file system's File system Monitor feature. This registration requests that the agent be notified when a client a requests a file open operation, prior to the file system's execution of the open operation" (col. 7, lines 59-63). This suggests the same concept of notifying when there is enough space for backup. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Dunphy, Koshisaka and Parthasarathy system to include an notification system as taught by Midgely in order to notify the user the space available to store the backup.

Conclusion

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Baoquoc N. To whose telephone number is at 571-272-4041, or unofficial fax number for the purpose of discussion (571) 273-4041 or via e-mail BaoquocN.To@uspto.gov. The examiner can normally be reached on Monday-Friday: 8:00 AM – 4:30 PM, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached at 571-272-4107.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks
Washington, D.C. 20231.

The fax numbers for the organization where this application or proceeding is assigned are as follow:

(571) 273-8300 [Official Communication]

/Baoquoc N To/

Primary Examiner, Art Unit 2162